



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/683,351	12/18/2001	Petra Leber	DE920000097US1	3040
46069	7590	02/24/2006	EXAMINER	
F. CHAU & ASSOCIATES, LLC 130 WOODBURY ROAD WOODBURY, NY 11797			RIZZUTO, KEVIN P	
			ART UNIT	PAPER NUMBER
			2183	

DATE MAILED: 02/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

FEB 24 2006

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 09/683,351

Filing Date: December 18, 2001

Appellant(s): LEBER ET AL.

Koon Hon Wong
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/16/2005 appealing from the Office action mailed 06/08/2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. The changes are as follows:

WITHDRAWN REJECTIONS

The following grounds of rejection are not presented for review on appeal because they have been withdrawn by the examiner. The 35 U.S.C. §112 second paragraph rejections of claims 1-10 have been withdrawn by the examiner.

GROUNDS OF REJECTION NOT ON REVIEW

The following grounds of rejection have not been withdrawn by the examiner, but they are not under review on appeal because they have not been presented for review in the appellant's brief. The 35 U.S.C. §103(a) Rejection of claim 6 has not been presented for review in the appellant's brief.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6493819 Mahurin et al. 12-2002

6449710 Isaman et al. 9-2002

Johnson, Mike. Superscalar Microprocessor Design. Prentice-Hall, Inc., 1991, pp. 133-134.

Hennessy, John L. and Patterson, David A. Computer Organization and Design, The Hardware/Software Interface. Morgan Kaufmann Publishers, Inc., 1998, pp. 118-119, 175, 185, 384, B-9 and the back inside cover

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1-5 and 7-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mahurin, U.S. Patent 6,493,819 in view of Isaman, U.S. Patent 6,449,710, and

Superscalar Microprocessor Design by Johnson and Computer Organization and Design , by Hennessy and Patterson.

Claim 1	
1. A method for operating a processor having an architecture of a larger bit-length with a program comprising instructions compiled to produce instruction results of at least one smaller bit-length, characterized by the steps of:	(The x86 instruction set has instructions capable of producing larger and smaller results. For instance, instructions can produce results of 8, 16, or 32 bits to be stored in registers of 32 bits (Column 1, line 63 to column 2, line 21 and column 7, lines 27-35). Therefore the processor that executes x86 instructions has an architecture of a larger bit length (32 bits) and also produces results of varying sizing, including "smaller" bit lengths (Also stated in Column 4, lines 58-64).
-Detecting when in program order a first instruction is to be dispatched which does not have a target register address as one	(Mahurin teaches the detection of x86 instructions that update portions of registers in the MROM unit. This includes

of its sources;	instances where instructions are detected that do not have the target address as its source (Column 5, lines 43-54 and column 10, lines 40-51)
-Wherein the first instruction is one of the instructions compiled to produce instruction results of at least one smaller bit-length;	(Abstract, Col. 1, line 63 to col. 2, line 21. The “first instruction” is an instruction that updates only a portion of a register, which is not given a special name by Mahurin, but Examiner refers to it as “the complex instruction” or “the smaller instruction” throughout the remainder of the document.)
-Adding an extract <i>operation</i> into an instruction stream before the first instruction, the extract operation comprising the following steps of;	(The MROM unit 34 adds a “read” (extract) operation into the instruction stream. A read of the destination register is done prior to the execution of the detected instruction. This read is a function performed by the processor as a result of control signals. The operation is carried out before the execution of the detected instruction, thus it was added prior to the instruction in the instruction stream

	(Column 5, 35-54)).
a. Dispatching instructions from an instruction queue into a Reservation Station;	(The MROM unit 34 conveys its instructions to the Decode Unit 20A, which in turn dispatches its instructions to the Reservation Station 22A. (Figure 1 and column 6, lines 24-52). Mahurin also teaches parallel dispatching to reservation stations. In figure 7 the multiple decode units all have the ability to dispatch instructions. Also, column 7, line 62 to column 8, line 26 describes the multiple instruction issuing per cycle.)
b. Issuing instructions to an Instructional Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme;	(Reservation station 22A issues instructions to an available execution unit (Functional unit 24A) once its source operands are available. Column 8, lines 27-46)
c. Executing instructions by an available IEU;	(column 8, lines 27-65)

Art Unit: 2183

<p>d. Setting an indication that the result of said extract operation needs to be written into the result field of the first instruction following the extract operation;</p>	<p>(Dest Size signal 210 indicates to Select Logic to write the result of the read (extract) operation into the result register of the detected smaller instruction. Figure 3 and column 14, line 52 to column 15, line 21.)</p>
<p>e. Writing the extract operation result into the result field of said first instruction, and into all registers of operands being dependent of said first instruction;</p>	<p>(The result of the read (extract) operation is merged with the results of the detected smaller instruction that only updates a portion of a register, and it is written into the destination register (result field) specified by the detected smaller instruction and the result is used by instructions that are dependent. Column 15, lines 14-40).</p>

2. While Mahurin does teach that a read of the destination register is done before the execution of the detected smaller instruction after the MROM unit detects the smaller instruction that does not have a target register as a source register, Mahurin **fails to teach that the read (extract) operation is a separate read instruction.** Therefore, Mahurin fails to teach an extract instruction is dispatched together with the following first instruction from an instruction queue into a reservation station, that the

extract instruction is issued to an Instructional Execution Unit (IEU) as soon as all source operand data is available and an IEU is available according to respective issue scheme, and that the extract instruction is executed by an available IEU.

3. However, as it has been shown above, Mahurin teaches that instructions from the MROM unit 34 are dispatched together from an instruction queue into a reservation station, that instructions are issued to an execution unit when all source operand data is available and when an execution unit is available and that the instructions are then executed by the available execution unit. This is important to note because an additional instruction added to the system (such as presented below) would inherently go through each of these steps for proper operation, as this is how the processor of Mahurin properly functions. Therefore, to summarize, the most significant difference in the references is that Mahurin teaches performing a “read operation” prior to the “first instruction” instead of inserting an actual, separate “extract instruction”. This inherently causes the additional limitations, that is, the dispatching, issuing, execution of said “extract instruction”, setting an indication that the result of *said extract instruction...*, and writing *the extract instruction* result to not be present in the teachings of Mahurin.

4. Isaman teaches the insertion of an extra “stitch” microinstruction (one instruction slot away from the detected instruction) in order to handle an instruction that updates only a portion of a register. The “stitch” microinstruction performs, among other things, an extraction of the data from the destination register, and it is processed like any other normal instruction (Figures 2 and 3). Specifically, figure 2 depicts the “stitch” instruction (instruction 100 #4A) being inserted into the instruction stream. This instruction is

inserted because instruction 100 #3 (ADD AL BL AL) updates only a portion of the destination register. The “stitch” instruction (instruction 100 #4A) reads the “EAX” register (destination register) and writes the contents retrieved back into the “EAX” register along with the results from the instruction 100 #3. The microinstruction is dispatched to the issue shelf, issued to an available execution unit and executed (Column 7, lines 45-65). Therefore, Isaman teaches using a separate instruction used to perform the read of the destination register when only operating on a portion of a register (the same problem addressed by Applicant’s disclosure and Mahurin).

5. It would have been obvious for one of ordinary skill in the art to recognize that a complex instruction, such as that taught in Mahurin, that requires a read function to be carried out as well as an arithmetic or logical operation, could be simplified by implementing a separate serialized read microinstruction that is inserted in an instruction slot one away from the detected instruction. Mahurin teaches that the read is carried out prior to the execution of the detected smaller instruction. Mahurin teaches that the MROM unit 34 detects the smaller instructions that only update a portion of a register. The separate microinstruction would come from the MROM unit 34 and be processed like any other instruction (or microinstruction), as Isaman teaches that the “stitch” instruction is inserted into the stream and is processed as any other normal instruction is. Mahurin teaches that instructions detected by the MROM unit are broken into simpler instructions and output to the Decode Units 20A-C (Column 5, lines 35-42, Mahurin). They are then dispatched to the Reservation Stations 22A-C. They are then issued to the execution units after their requirements are met, which is explained in

Column 8, lines 27-46 of Mahurin. The requirements for issuing that Mahurin lays out would require that the separate read microinstruction would have to be issued and executed prior to the smaller instruction. (“A read of the destination is done prior to execution of the instruction”, Mahurin, abstract, lines 11-12, col. 5, lines 44-48.) In turning the read operation into a separate instruction to be dispatched, issued, and executed (as taught by Isaman), instead of concurrently doing the read while the detected smaller instruction is being processed (as taught by Mahurin), a hardware reduction can occur. A read of a register, and write to another register is done on many instructions in the x86 instruction set, therefore, the hardware needed to implement a read microinstruction is already present. Breaking down one complex instruction into multiple simpler instructions is well known in the art and is taught in Hennessy and Patterson, first paragraph, page 175, which is copied below.

“Designers of instruction sets sometimes provide more powerful operations than those found in MIPS. The goal is generally to reduce the number of instructions executed by a program. The danger is that this reduction can occur at the cost of simplicity, increasing the time a program takes to execute because the instructions are slower. This slowness may be the result of a slower clock cycle time or of requiring more clock cycles than a simpler sequence (see section 2.8 on page 82).

The path toward operation complexity is thus fraught with peril. To avoid these problems, designers have moved toward simpler instructions. Section 3.313 demonstrates the pitfalls of complexity.”

6. Furthermore, the additional page cited by Hennessy and Patterson above further supports this well known principle of simplicity versus complexity in instructions.

Section 3.13, page 185, states:

“Fallacies and Pitfalls

Fallacy: More powerful instructions mean higher performance.

Part of the power of the Intel 80x86 is the prefixes that can modify the execution of the following instruction. One prefix can repeat the following instruction until a counter counts down to 0. Thus, to move data in memory, it would seem that the natural instruction sequence is to use move with the repeat prefix to perform 32-bit memory-to-memory moves. On a 133-MHz Pentium (with the Triton chip set, 60-ns EDO DRAM, 256-KB cache), this user-level program can move data at about 50 MB/sec.

An alternative method, which uses the standard instructions found in all computers, is to load the data into the registers and then store the registers back to memory. This second version of this program, with the code replicated so as to reduce loop overhead, copies at about 50 MB/sec on the same machine or 1.5 times faster. A third version, which used the larger floating-point registers instead of the integer registers of the 80xx86, copies at about 80 MB/sec, or 2.0 times faster than the complex instruction."

7. Furthermore, breaking down complex instruction into multiple, simpler microinstructions are the general intention of an MROM unit anyway. The smaller instruction would only need to be detected once (in the MROM unit) and the reservation station would not need to implement its special issuing scheme (described in column 11, lines 21-33). The special issuing scheme requires extra hardware such that a third operand, holding the contents of the prior read of the destination register, can be held in the reservation station entry for the one complex (smaller) instruction. The special issuing scheme further requires the complex (smaller) instruction to wait for the retrieval of the third operand. (Col. 11, lines 21-33)

8. The reduction and simplification of hardware, as well as there no longer being a need for the complex instruction to wait for the third operand, would have provided the motivation to implement the read instruction of Mahurin in a separate microinstruction inserted into the instruction stream by the MROM unit 34. Furthermore, as previously cited above, Hennessy and Patterson teach that designers have moved toward simpler

instructions as operation complexity is “fraught with peril.” This would have further provided the motivation to one of ordinary skill in the art to break down the single complex instruction into two instructions, as described above.

9. However, Mahurin, in view of Isaman, does not teach the dispatching of the read instruction and the detected first instruction together to a single reservation station. Mahurin teaches that multiple instructions are dispatched together, however it is to multiple reservation stations instead of one reservation station.

10. Johnson teaches a single reservation station (central window) that can have multiple instructions dispatched to it. A single reservation station is more efficient than multiple reservation stations because it holds all instructions for issue regardless of which functional units execute the instructions (page 133 of Johnson). It would have been obvious to combine the use of only one reservation station with the parallel dispatching of Mahurin, which would cause parallel dispatching to a single reservation station. When both instructions are decoded and known, and there is room in the reservation station, they would be dispatched together, the read instruction to be executed first and the detected smaller instruction executed after. By implementing only one reservation station, the multiple dispatching of instructions that are in order (column 6, lines 24-38 of Mahurin) would cause the read instruction and the detected smaller instruction to be dispatched together to a reservation station, since they are back to back as taught in figure 2 of Isaman.

11. Reducing the number of reservation stations to one would have allowed the instructions to be dispatched together to a reservation station. The increased efficiency

of one reservation station would have provided the motivation to one of ordinary skill in the art to do the reduction.

12. **As per independent claims 8, 9, and 10,** given the similarities between claim 1 and claims 8-10, the arguments as stated for the rejection of claim 1 also apply to claims 8-10.

13. **As per claim 2,** Mahurin, in view of Isaman and, teaches the method according to claim 1 including the step of writing the extract instruction result into the result field of said first instruction. It is taught that the extract and detected smaller instruction are one after the other sequentially. It is also taught that a reservation station addresses registers for instructions by tag addresses (Column 7, lines 36-61 of Mahurin). However, Mahurin, in view of Isaman and Johnson, is silent on how the writing of the result of the extract instruction is controlled. Hennessy and Patterson teach a program counter in which the next sequential address is determined by incrementing it (Page 384). A tag specifies a location and therefore is just another term for an address.

14. It would have been obvious to one of ordinary skill in the art that when the two sequential instructions are dispatched, as taught above, where the first read (extract) instruction must write to the destination register of the instruction following it (smaller detected instruction), the tag should be incremented to whatever the tag specifying the destination register is. This is the same as the program counter of Hennessy and Patterson being incremented by whatever value is necessary to get to the next

sequential instruction. Incrementing a program counter to get a new address value is used by modern processors and therefore incrementing the value of a tag (address) is very well known in the art. Using a well-known method in order to allow two instructions writing to the same renamed register would make the design of the processor simpler and easier. This would have provided the motivation to combine the incrementing method of the program counter of Hennessey and Patterson with Mahurin in view of Isaman and Johnson.

15. **As per claim 3,** Mahurin, in view of Isaman and Johnson, teaches the method according to claim 1, and also teaches having the larger bit length equal to 32 bits and having a smaller bit length equal to 16 bits.

16. Mahurin fails to teach that the larger bit length is 64-bits and the smaller bit length is 32-bits.

17. It would have been obvious for Mahurin to have a larger bit length of 64-bits and a smaller bit length of 32-bits because more data can be stored in registers of those sizes than the current register sizes. This would allow more precise values to be stored in a single register and allow more data overall to be present inside a register file with the same size bus for addressing registers.

18. It also would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Mahurin's invention on a larger bit length equal to 64-bits and a smaller bit length equal to 32-bits since it has been held that changes in size

that have no unexpected results is obvious. In re Rose, 220 F.2d 459, 105 USPQ 237 (CCPA 1955).

19. **As per claim 4**, Mahurin, in view of Isaman and Johnson, teaches the method according to claim 1, and also teaches having the larger bit length equal to 32 bits and having a smaller bit length equal to 16 bits.

20. Mahurin fails to teach that the larger bit length is 128-bits and the smaller bit length is 64 or 32-bits.

21. It would have been obvious for Mahurin to have a larger bit length of 128-bits and a smaller bit length of 64 or 32-bits because more data can be stored in registers of those sizes than the current register sizes. This would allow more precise values to be stored in a single register and allow more data overall to be present inside a register file with the same size bus for addressing registers.

22. It also would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Mahurin's invention on a larger bit length equal to 128-bits and a smaller bit length equal to 64 or 32-bits since it has been held that changes in size that have no unexpected results is obvious. In re Rose, 220 F.2d 459, 105 USPQ 237 (CCPA 1955).

Claim 5	
5. The method according to claim 1, further comprising the step of when said	(Mahurin teaches the selective insertion of extract instructions depending on if the

second instruction is dependent of the first instruction, selectively inserting an extract instruction:	result of an instruction updates only a portion of a register or not (Column 5, lines 43-54). This selective insertion occurs for all instructions going through the MROM unit 34, including when a second instruction is dependent on the detected first instruction.)
Claim 7	
7. The method according to claim 1, further comprising the step of associating the same instruction execution unit for said first and said extract instruction:	(Figure 3, the results of the read taught by Mahurin and the detected first instruction are both associated with the same (IEU) functional unit 24A. Mahurin teaches one functional unit is used with both of their instruction results, which shows the two instructions are associated with the same functional unit by means of their results).

(10) Response to Argument

1. On page 10, Applicant argues:

"(i). The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest 'adding an extract instruction into an instruction stream before the first instruction,' as claimed in claims 1, 8, 9 and 10.

The Examiner contends that col. 5, lines 35-54 of Mahurin teaches 'adding an extract instruction into an instruction stream before the first instruction.' (Paper no. 20040907, p.5)." Col. 5, lines 35-42 of Mahurin teaches the following:

MROM instructions are instructions which are determined to be too complex for decode by decode units 20. MROM instructions are executed by invoking MROM unit 34. More specifically, when an MROM instruction is encountered MROM unit 34 parses and issues the instruction into a subset of defined fast path instructions to effectuate the desired operation. MROM unit 34 dispatches the subset of fast path instructions to decode units 20.

Col. 5, lines 44-48 of Mahurin further teaches "[w]hen MROM unit 34 detects such an instruction and the portion of the destination which is not updated by the instruction is detected as being required by the instruction, a read of the destination will be done prior to execution of the instruction." The Examiner refers to this 'read of the destination' as a 'read operation.' The Examiner assumes that the MROM unit 34 adds the read operation into the instruction stream prior to an execution instruction. Applicants respectfully submit that the Examiner's assumption is without merit.

As quoted above, the recited portion of Mahurin states that the 'MROM unit 34 dispatches the subset of fast path instructions to decode units 20.' Mahurin does not teach that the MROM unit 34 dispatches anything other than the subset of fast path instructions. Although the recited portion of Mahurin does state that a 'read of the destination will be done prior to the execution of the instruction,' Mahurin does not address how or where the read operation is performed or what performs the read operation."

2. This is not found persuasive for the following reasons:
 - a. To begin, the quoted portion of the Office Action (Paper no. 20040907, p.5) Applicant has cited is incorrect and is a misrepresentation of the Examiner's position. Furthermore, the first Office Action mailed in regards to this application was mailed 11/8/2004, two months after the citation Applicant has given. Examiner assumes Applicant was referring to the Final Office Action mailed 6/8/05, as Page 5 of the Final Action (6/8/05) contains a *similar* line, which states, "[Mahurin teaches] adding an

extract operation into an instruction stream before the first instruction."

(Emphasis added by Examiner to indicate the possibly misquoted portion)

b. Examiner contends that Mahurin does not teach a separate extract *instruction* being inserted into the instruction stream. Rather, Mahurin teaches an extract operation is performed as part of the processing of the complex, smaller instruction that updates only a portion of a register.

Continuing on from Final Action (6/8/05), page 5, it was stated, "A read of the destination register is done prior to the execution of the detected instruction. This read is a function performed by the processor as a result of control signals. The operation is carried out before the execution of the detected instruction so it was added prior to the instruction in the instruction stream. (Column 5, [lines] 35-54)." Furthermore, Examiner more explicitly stated the position that a separate "extract instruction" is not inserted into the instruction stream throughout prosecution, for example:

- i. "Mahurin fails to teach that the read (extract) operation is a separate read instruction..." (From Final Office Action, (6/8/05), page 6.)
- ii. "As stated in the previously set forth 35 U.S.C. 103 rejection, what Mahurin does not teach, is the "read operation" being implemented as a separate instruction. Instead, it is merely another function performed for the complex instruction." (From Advisory Action, mailed 08/30/2005.)

c. Therefore, as has been shown above, it has been made clear that the Examiner *does not assume* a separate extract (read) instruction is inserted into the instruction stream.

d. Furthermore, Mahurin teaches that the MROM unit 34 receives complex instructions and “parses and issues the instruction into a subset of defined fast path instructions to effectuate the desired operation”, and that the “MROM unit 34 detects when an instruction updates only a portion of a destination” and as a result, “a read of the destination will be done prior to execution of the instruction.” [Col. 5, lines 38-48.] Also, Applicant has stated, and Examiner agrees, “Mahurin does not teach that the MROM unit 34 dispatches anything other than the subset of fast path instructions.” (Page 10, final paragraph.) After the MROM unit 34 detects an instruction updates only a portion of a destination register will be updated, it outputs an instruction that goes to the Decode unit and causes the Decode unit to output the read request. (Col. 14, lines 27-39 and fig. 1 depicts the output of the MROM unit 34 going to *the Decode unit 20A*). Thus, Mahurin teaches the MROM unit 34 causes the read operation to occur, which inherently is a result of control signals coming out of the MROM unit 34, as the MROM unit 34 is a hardware component in a processing system. Furthermore, it is unclear why Mahurin would state “**When MROM unit 34 detects** such an instruction and the portion of the destination which is not updated by the instruction is detected as being required by the instruction, **a read of the destination will be done** prior to execution of the instruction” (Col. 5, lines 44-48) if the MROM did not in some manner cause the read to occur. Therefore, Examiner has

adequately supported the statements of page 5, Final Action (6/8/05), that is, that “[Mahurin teaches] adding an extract *operation* into an instruction stream before the first instruction.”

3. On page 11, Applicant argues:

“It should further be noted that Mahurin teaches that the ‘read operation’ is done only conditionally. Col. 5, lines 50-54 of Mahurin teaches that ‘when MROM unit detects an instruction which updates only a portion of a destination and the portion of the destination which is not updated by the instruction is detected as not being required by the instruction, [then] a read of the destination is not done.’ Therefore, even assuming, arguendo, that Mahurin teaches ‘adding an extract instruction into an instruction stream before the first instruction,’ the present claims do not restrict this step to the conditional taught by Mahurin.”

4. This is not found persuasive for the following reasons:

e. Applicant appears to be relying upon features (i.e., that the read operation is not to be done conditionally) that are not recited in the rejected claim(s).

Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

f. Applicant appears to be arguing that Mahurin teaches a processor that performs the claimed invention in certain instances (i.e., conditionally). Thus Mahurin teaches the claimed method steps in question, since they occur in certain instances. Merely because the claimed method step does not occur in every possible instance would not preclude Mahurin from being applied to the limitation in question. According to the MPEP, “comprising” language is open-

ended (see cited portion below), and thus, further limitations present in Mahurin do not prevent the teachings from being applied as prior art.

From MPEP 2111.03 [R-3] Transitional Phrases

The transitional phrases "comprising", "consisting essentially of" and "consisting of" define the scope of a claim with respect to what unrecited additional components or steps, if any, are excluded from the scope of the claim.

The transitional term "comprising", which is synonymous with "including," "containing," or "characterized by," is inclusive or open-ended and does not exclude additional, unrecited elements or method steps. See, e.g., > Mars Inc. v. H.J. Heinz Co., 377 F.3d 1369, 1376, 71 USPQ2d 1837, 1843 (Fed. Cir. 2004) ("like the term comprising,' the terms containing' and mixture' are open-ended.");< Invitrogen Corp. v. Biocrest Mfg., L.P., 327 F.3d 1364, 1368, 66 USPQ2d 1631, 1634 (Fed. Cir. 2003) ("The transition comprising' in a method claim indicates that the claim is open-ended and allows for additional steps."); Genentech, Inc. v. Chiron Corp., 112 F.3d 495, 501, 42 USPQ2d 1608, 1613 (Fed. Cir. 1997) ("Comprising" is a term of art used in claim language which means that the named elements are essential, but other elements may be added and still form a construct within the scope of the claim.); Moleculon Research Corp. v. CBS, Inc., 793 F.2d 1261, 229 USPQ 805 (Fed. Cir. 1986); In re Baxter, 656 F.2d 679, 686, 210 USPQ 795, 803 (CCPA 1981); Ex parte Davis, 80 USPQ 448, 450 (Bd. App. 1948) ("comprising" leaves "the claim open for the inclusion of unspecified ingredients even in major amounts"). >In Gillette Co. v. Energizer Holdings Inc., 405 F.3d 1367, 1371-73, 74 USPQ2d 1586, 1589-91 (Fed. Cir. 2005), the court held that a claim to "a safety razor blade unit comprising a guard, a cap, and a group of first, second, and third blades" encompasses razors with more than three blades because the transitional phrase "comprising" in the preamble and the phrase "group of" are presumptively open-ended. "The word comprising' transitioning from the preamble to the body signals that the entire claim is presumptively open-ended." Id. In contrast, the court noted the phrase "group consisting of" is a closed term, which is often used in claim drafting to signal a "Markush group" that is by its nature closed. Id. The court also emphasized that reference to "first," "second," and "third" blades in the claim was not used to show a serial or numerical limitation but instead was used to distinguish or identify the various members of the group. Id.<

5. Applicant argues on page 11, second paragraph:

Art Unit: 2183

"Throughout all of his rejections, the Examiner makes the flawed assumption that in Mahurin, the 'read operation' is an 'instruction' executed by the functional units 24 of Figure 1. That is, when Mahurin refers to 'instructions,' the Examiner assumes that this necessarily includes the 'read operation.' At no point does Mahurin refer to the 'read operation' as an 'instruction.' At no point does Mahurin even state the function un[i]ts 24 perform the 'read operation.' Indeed, that Mahurin does not refer to the 'read operation' as an 'instruction' strongly indicates that Mahurin intends for the 'read operation' to not be an 'instruction.' Therefore, the Examiner cannot simply interpret the 'read operation' as an 'instruction' in hindsight. Such an interpretation is clearly improper."

6. This is not found persuasive for the following reasons:
 - g. These statements of Examiner's position are in direct contrast to an explicit position put forth in each Office Action. Examiner does not contend that Mahurin teaches "the 'read operation' is an 'instruction' executed by the functional units 34 of Figure 1. No such assumption has been made. The following our direct quotes from previous actions mailed:
 - iii. "Mahurin fails to teach that the read (extract) operation is a separate read instruction..." (From Final Office Action, (6/8/05), page 6.)
 - iv. "As stated in the previously set forth 35 U.S.C. 103 rejection, what Mahurin does not teach, is the "read operation" being implemented as a separate instruction. Instead, it is merely another function performed for the complex instruction." (From Advisory Action, mailed 08/30/2005.)
 - h. Furthermore, it is unclear what the purpose of the combination of Isaman and Mahurin would be if such an assumption was made. The Final Office Action mailed 6/8/05 states on pages 7-8, paragraphs 13-15:
 - v. "Isaman teaches the insertion of an extra microinstruction (one instruction slot away from the detected instruction) in order to handle an

instruction that updates only a portion of a register...it would have been obvious...that an instruction that requires a read function to be carried out could be simplified by implementing a separate serialized read microinstruction that is inserted in an instruction slot one away from the detected instruction...The reduction and simplification of hardware would have provided the motivation to implement the read instruction of Mahurin in a separate microinstruction inserted into the instruction stream by the MROM unit 34."

- i. If Examiner had already made the assumption that the MROM unit 34 inserts a separate extract *instruction* into the instruction stream, there would be no reason for the 103 combination of Mahurin, in view of Isaman, set forth, since the teaching of the combination as presented by the Examiner is clearly to add a separate extract instruction.

7. Applicant argues on page 12:

"(ii). The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest 'setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction,' as claimed in claims 1, 8, 9 and 10.

The Examiner contends that Figure 3 and col. 14, lines 52 to column 15, line 21 of Mahurin teaches 'setting an indication that the result of said extract instruction needs to be written into a result field of the first instruction following the extract instruction.["] More particularly, the Examiner contends that 'Dest Size signal 210 indicates to Select Logic to write the result of the read (extract) operation into the result register of the detected smaller instruction.' Applicants respectfully disagree.

Col. 15, lines 14-21 of Mahurin states the following regarding select logic 202: Select logic 202 provides select signals 240, 242, and 244 to multiplexors 204, 206 and 208 respectively. *Based on the destination size and location received upon bus 210, the select signals will determine which portions of the data fed to each multiplexor will be gated out.* It is this gating out of data from multiplexors 204, 206 and 208 which will form the final result to be conveyed from functional unit 24A upon bus 38a.

That is, the bus 210 provides only destination size and location information to the select logic 202. Mahurin does not teach that any information regarding result of the 'read operation' (which the Examiner contends is the claimed 'extract instruction') is sent of the select logic 202. Further, Mahurin does not indicate that the executed instruction (which the Examiner contends is the claimed 'first instruction') even contains a 'result field' as claimed in the instant claims. Therefore, the Examiner has no basis to assume that 'Dest Size signal 210 indicates to Select Logic to *write the result of the read (extract) operation* into the result register of the detected smaller instruction."

- j. This is not found persuasive for the following reasons:
 - vi. Examiner cited Fig. 3, and col. 14, lines 52 to column 15, line 21 in response to the limitation in question. Furthermore, Examiner stated, "Dest Size signal 210 indicates to Select Logic to write the result of the read (extract) operation into the result register of the detected smaller instruction." (Final Office Action, mailed 6/8/05, page 6, paragraph d) Mahurin, in col. 14, lines 55-57, states, "When only a portion of the destination register is being updated, the contents of the destination register are conveyed upon bus 212." The "contents of the destination register" is the data from the prior "read operation." This is supported on col. 11, lines 21-33 and col. 5, lines 43-50. Figure 3 depicts the Destination Bus 212 which carries the prior read operation data, and it is sent to Muxes A, B and C (204, 206 and 208 respectively). From the above cited portion of Mahurin (Col. 15, lines 14-21), "Based on the destination size and location received upon bus 210, the select signals will determine which portions of the data fed to each multiplexor will be gated out." The results are then sent on Result Bus 38a back to the Reorder

Buffer to the destination register for the detected, smaller instruction, i.e., "a result field of the first instruction." (Supported in fig. 2 and col. 9, lines 14-23). Therefore, the Dest Size signal 210 sends an indication in the Select Logic 202, so that the Select Logic 202 can indicate to the Muxes 204, 206 and 208, to write the result of the prior destination register read (found on Destination Bus 212) to the result field of the "first instruction."

8. Applicant argues on pages 13-14:

"“(iii). The combination of Mahurin, Isaman, Johnson and Hennessy fails to teach or suggest “writing the extract instruction result into the result field of said first instruction, and into all fields of operands being dependent of said first instruction,’ as claimed in claims 1, 8, 9 and 10.

The Examiner contends that col. 15, lines 14-40 of Mahurin teach 'writing the extract instruction result into the result field of said instruction.' Applicants respectfully disagree.

The recited portion of Mahurin teaches the merging of the result of instruction execution with 'previous data,' which is unrelated to the claimed invention. The recited portion of Mahurin does not refer to the read operation (which the Examiner contends is the claimed 'extract instruction') at all. Further, the recited portion of Mahurin does not teach that the executed instruction (which the Examiner contends is the claimed 'first instruction') contains a 'result field' as claimed in the instant application. Therefore, it is unclear exactly what the Examiner is contending is being written into.

- k. This is not found persuasive for the following reasons:
- vii. As Applicant states, "The recited portion of Mahurin teaches the merging of the result of instruction execution with 'previous data'". The 'previous data' is the result of the read of the destination register prior to the execution of the first, detected, smaller instruction. While the cited portion of the reference does not explicitly state that the 'previous data' is

the result of the prior read of the destination register, it is absolutely clear from teachings throughout the reference, including the abstract, that this is the case. Furthermore, the “previous data” is clearly related to the invention, as it is the extracted data from the read operation. Examiner was merely attempting to specifically point out where Mahurin teaches the writing of the result into the result field and into fields of dependent operations. However, to support the assertion that the ‘previous data’ that is merged is the resultant data from the read operation, portions of Mahurin are cited below:

(1) “In one embodiment, MROM unit 34 detects when an instruction updates only a portion of a destination. When MROM unit 34 detects such an instruction and the portion of the destination which is not updated by the instruction is detected as being required by the instruction, a read of the destination will be done prior to execution of the instruction. The destination contents will subsequently be merged with the instruction result.” (Col. 5, lines 43-50)

(2) “If only a portion of the destination is updated by the instruction, a read of the destination is done prior to execution of the instruction and the data read from the destination is merged with the results of the instruction execution.”

(Abstract)

viii. Mahurin further teaches that after the result of the read of the destination with the instruction result are merged, they are forwarded upon result bus 38a (figs. 2 and 3) onto the Reorder Buffer 32, to be stored in the result register. (Supported in col. 9, lines 14-23) Furthermore, the result bus 38A has its data, along with its corresponding tag, sent to the reservation station when the results are also sources for upcoming instructions (see fig. 2, col. 12, lines 4-26), as indicated by the tags of the sources in the reservation station.

9. Applicant argues on page 14:

"Also, as previously noted, the 'read operation' of Mahurin is performed only if the MROM unit 34 meets a conditional statement. This conditional is not present in the instant claims. Mahurin also teaches that when the read is performed, the contents result from the read 'will subsequently be merged with the instruction result.' (Mahurin, col. 5, lines 47-48) This indicates that after the 'read operation' is performed, the contends resulting from the read operation are not kept. Therefore, if Mahurin does not even keep the contents resulting from the read, it does not follow that such contents are written into 'the result field of said first instruction,' as claimed in claims 1, 8, 9, and 10."

I. This is not found persuasive for the following reasons:

ix. Applicant's deduction, that is, since Mahurin teaches the contents resulting from the read will subsequently be merged with the instruction result, "indicates that after the read operation is performed, the contents resulting from the read operation are not kept", is incorrect. The merged instruction results are clearly stored in the associated instruction result field in the reorder buffer and also forwarded to the reservation station. As

stated in the abstract, the point of the invention is to handle when an instruction updates only a portion of a destination register. To handle this situation, a prior read of the destination occurs, the result of the read and the result of the instruction **are merged**, and the **data is stored** in the correct destination. Any other interpretation is clearly contradictory to the teachings of Mahurin. Therefore, since the result of the read is merged, **then stored**, the results of the read are clearly “kept.” Additionally, since the results of the read are merged with the results of the detected, smaller instruction, then stored in the destination of the detected, smaller instruction, the results of the read are stored in a “result field of the first instruction.” Furthermore, *the results forwarded on result bus 38A contain a tag so as to be provided to all instructions that are dependent and are waiting for the result as a source operand in the reservation station 22.* (Col. 12, lines 4-29).

10. Applicant argues on page 15:

“(iv). The only suggestion for selectively piecing together the features of disparate elements in Mahurin, Isaman, Johnson and Hennessy, in the manner proposed by the Examiner stems from hindsight knowledge impermissibly derived from Appellants' disclosure.

To support combining Mahurin, Isaman, Johnson and Hennessy, the Examiner makes numerous unsupported assumptions with regards to the references and relies on disparate elements from various references.

- m. This is not found persuasive for the following reasons:

x. As addressed above, each “unsupported assumption” Applicant refers to was either a misrepresentation of Examiner’s position or fully supported by the teachings in Mahurin.

11. Applicant arguments continue on page 15:

“It should be noted that the Examiner’s motivations to combine are artificially created without any objected teaching in the prior art. As such, the only suggestion for selectively piecing together the features of these disparate elements in the manner proposed by the Examiner stems from hindsight knowledge impermissibly derived from the Appellants’ disclosure.”

- n. This is not found persuasive for the following reasons:
- xi. Applicant has yet to address the motivation specifically set forth in each Office Action except in general terms such as present above. Additionally, the claim that the motivations to combine “are artificially created without any objected teachings in the prior art” is merely an allegation. Looking at the Office Actions set forth, motivation was clearly presented as to why one of ordinary skill in the art would combine the references. Below are examples of such:

xii. “It would have been obvious for one of ordinary skill in the art to recognize that an instruction that requires a read function to be carried out could be simplified by implementing a separate serialized read microinstruction that is inserted in an instruction slot one away from the detected instruction...The requirements for issuing that Mahurin lays out would require that the separate read microinstruction would have to be issued and executed prior to the smaller instruction. In turning the read operation into a separate instruction to be dispatched, issued, and executed, instead of concurrently doing the read while the detected smaller instruction is being processed, a hardware reduction can occur... **Breaking down one complex instruction into multiple simpler instructions is well known in the art and is taught in Hennessy and Patterson, page**

175 and is the intention of the MROM unit. The smaller instruction would only have to be detected once (in the MROM unit) and the reservation station would not need to implement its special issuing scheme (described in column 11, lines 21-33). The hardware could then be simplified and reduced in the reservation station.

xiii. The reduction and simplification of hardware would have provided the motivation to implement the read instruction of Mahurin in a separate microinstruction inserted into the instruction stream by the MROM unit 34." Final Office Action (6/8/05) (**Emphasis added by Examiner to indicate a specific example of an objective teaching in the prior art that was used as motivation for the combination**)

12. Applicant argues on page 15:

"It should be noted that Isaman teaches exactly what is described in the instant application in Figure 6. Isaman simply replaces the 'merge' instruction with the 'stitch' instruction. However, the use of such a 'stitch' instruction is expressly rejected by the instant application because of performance loss introduced by dependencies between the smaller bit length instruction, the merge (or stitch) instruction and the instruction consuming the target register."

- o. This is not found persuasive for the following reasons:
 - xiv. Examiner directs Applicant's attention to the 35 U.S.C. 103 rejection previously set forth. Mahurin teaches wherein the read of the destination register is done before the execution of the detected smaller instruction. [Mahurin, Col. 5, lines 35-54]. Therefore, Mahurin teaches a system that does not execute a "merge" or "stitch" operation or instruction after the smaller result has been produced. As stated in the previously set forth 35 U.S.C. 103 rejection, what Mahurin does not teach, is the "read operation" being implemented as a separate instruction. Instead, it is merely another function performed for the complex instruction. Isaman,

however, teaches wherein the “read operation” is performed explicitly by a separate instruction, i.e., the stitch instruction. The aspect of Isaman that is being combined with Mahurin is the fact that the read operation should be implemented as a separate instruction; it should not be a requirement for the first (smaller) instruction to perform the read. This will convert the first instruction into a simpler instruction, thus reducing the hardware, as stated in the 35 U.S.C. 103 rejection.

xv. Examiner acknowledges that Isaman teaches a similar solution to the admitted prior art by Applicant, however, Examiner also notes that the admitted prior art would also provide teachings to modify Mahurin in the manner Isaman does. To reiterate, the only substantial aspect of Isaman that is being combined with Mahurin, is that a separate instruction is dispatched, issued and executed in order to perform the read of the destination register, and motivation has been provided to do so.

xvi. Furthermore, the “flaws” of the Isaman-like teachings, as described by Applicant in the specification, have been addressed by Mahurin. Mahurin teaches wherein the read of the destination register is not done subsequently to the first (smaller) instruction. The read operation occurs prior to the first (smaller) instruction’s execution. This specifically overcomes the “flaws” Applicant has laid out in regard to the prior art of figure 6, that is, Applicant states, “the drawback of the solution is that due to the execution of the merge instruction at least one extra cycle is added

Art Unit: 2183

before the R1 data is available" (because it is executed subsequently to the first (smaller) instruction) [Spec. page 8, paragraph 45].

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

February 1, 2006
Kevin Rizzuto

Conferees:

Eddie Chan
SPE

Donald Sparks
SPE

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

DONALD SPARKS
SUPERVISORY PATENT EXAMINER
